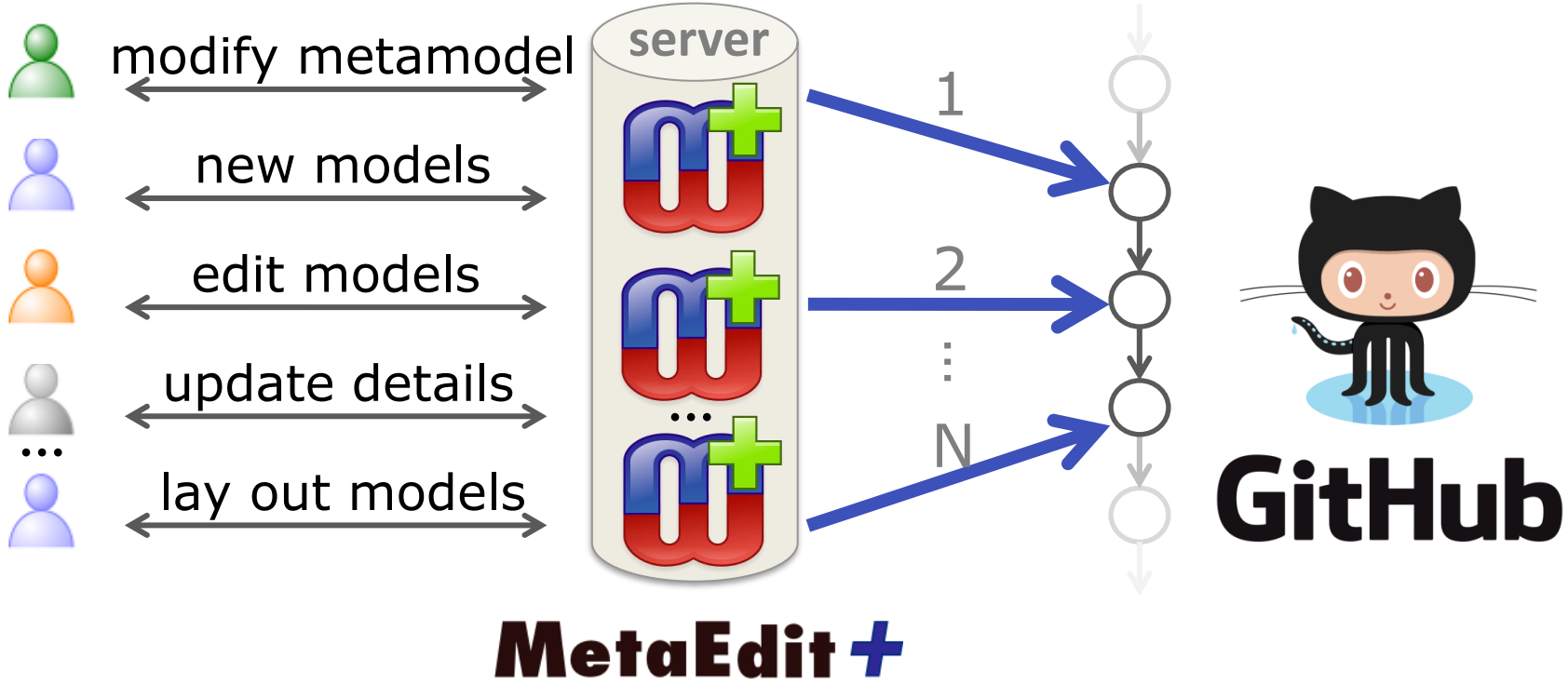
Collaborative Modeling and Metamodeling in the Cloud with Versioning

Steven Kelly, Juha-Pekka Tolvanen
stevek | jpt @metacase.com

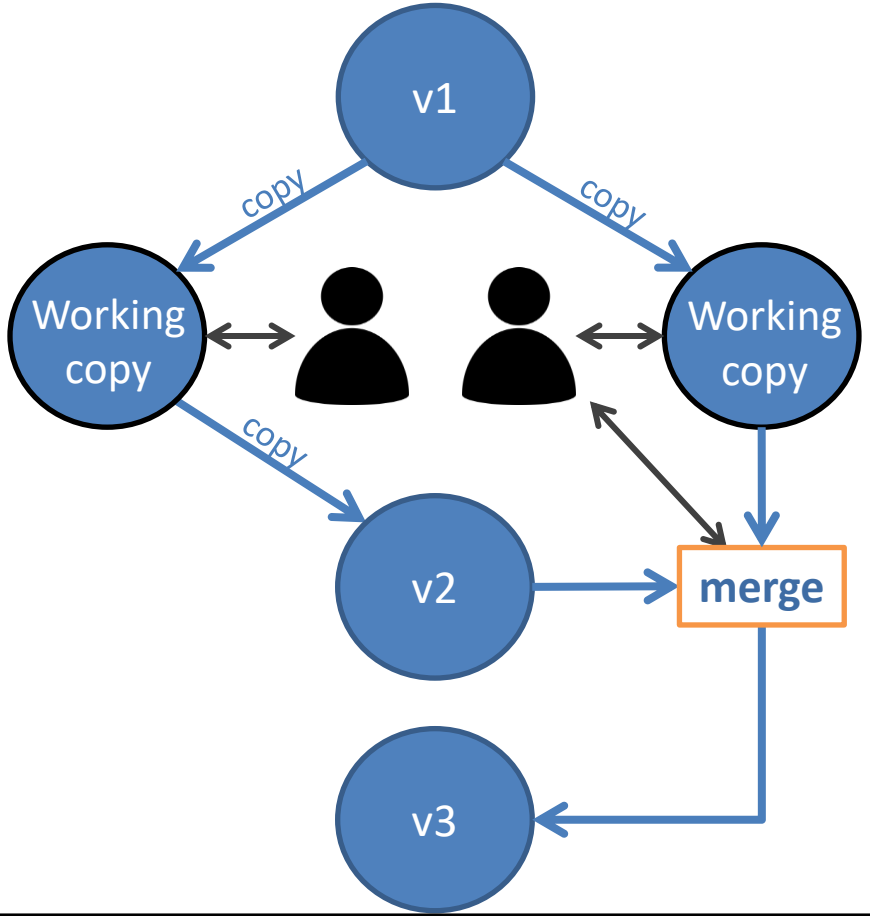
Collaborative (meta-)Modeling and Versioning of Models and Languages

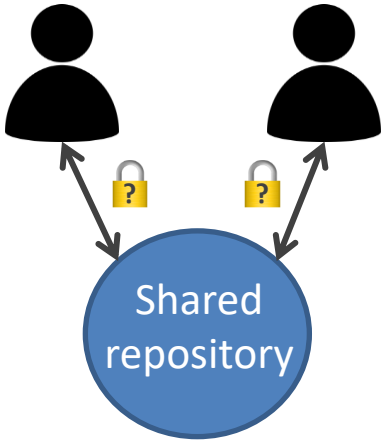


Clone *vs.* Share

merge after the fact

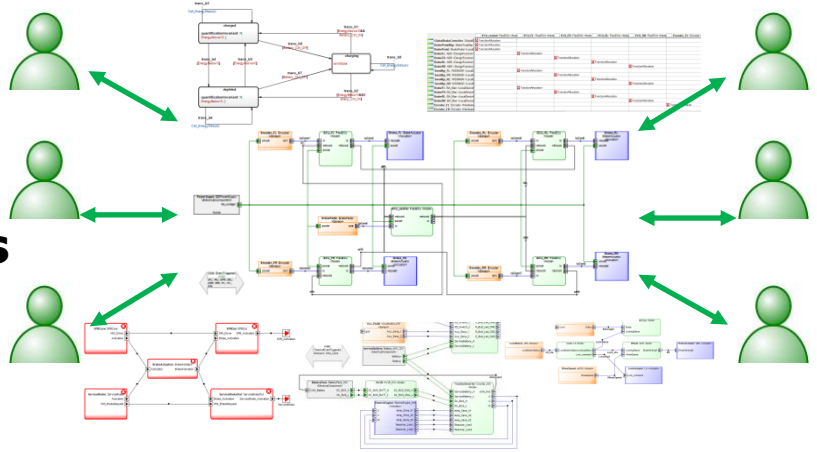
continuous integration





MetaEdit+

- Mature, commercial, supported Language Workbench
- Collaborative modeling support
  - Multiple developers
  - Multiple models
  - Multiple languages
  - **Tracks history & changes**
  - **Integrated with VCSs**
- Collaborative language engineering
  - **Updates existing models**
  - **Integrates with VCSs**



Collaboration at the object level

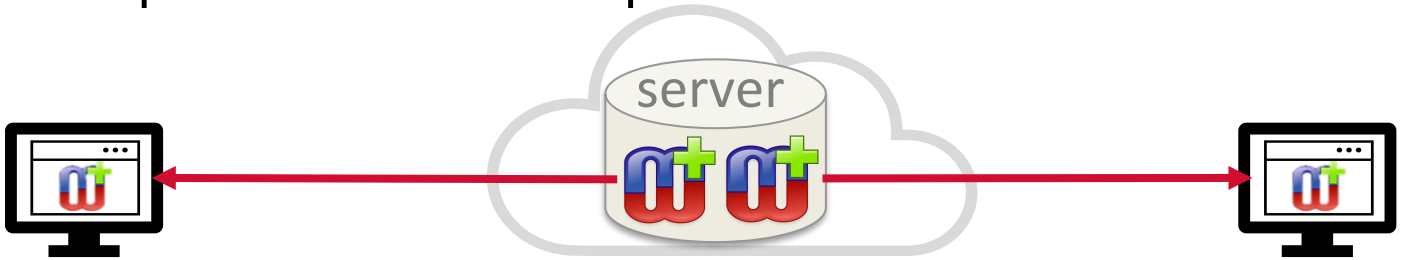
- Repository-based: users work in parallel on same data
  - No need to merge
  - Models and metamodels can be edited simultaneously
- Locks ensure no conflicts
  - Fine granularity, so can work closely without lockout
- Low ceremony: locking and unlocking is automatic
  - Users can focus on work, not tool
- Design transactions: minutes to hours
  - Commit a coherent set of changes together
    - Releasing half-finished/inconsistent data would confuse others
  - See a consistent version of repository during transaction
    - Avoid being distracted by others' changes appearing live

RSL: Remote Shared License

- IT departments want to avoid installing software
- Want to allow users to work from anywhere

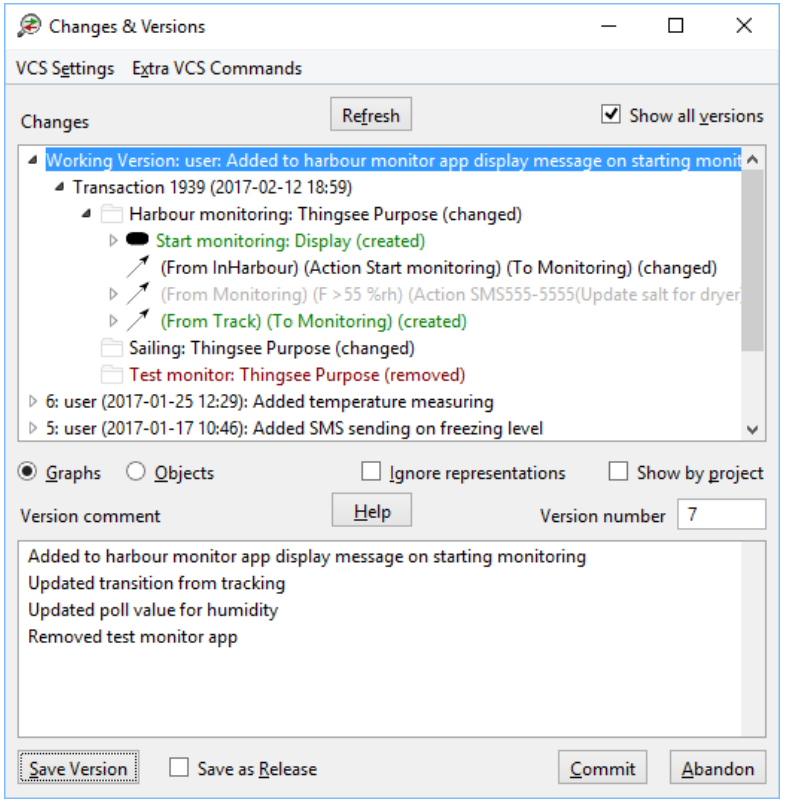
- Graphical modelling has high performance reqs at client
- Large repositories are slow to move over a network
  - Particularly with high latency over VPNs + load as needed

- Put MetaEdit+ server and client on one Windows Server
- Multiple remote desktop connections from users

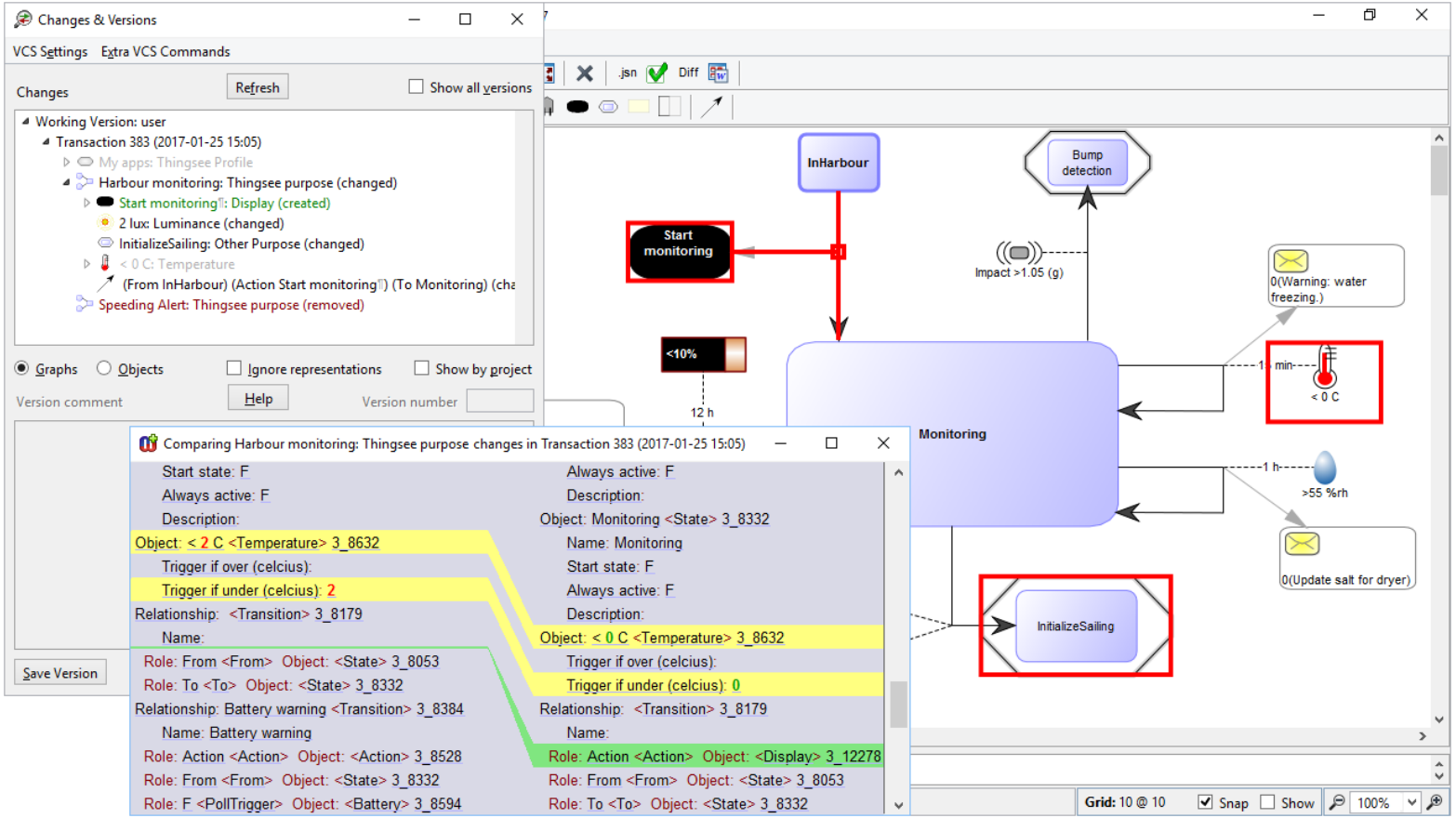


Changes and Versions tools

- Automatic trace of model changes
- See changes graphically directly in your models
- View changes as a tree using your language's structure and symbols, not XML
- Compare changes as a textual diff with live links to models
- Filters (your changes/all changes, data/representation)
- All the info you need, clearly: => easy to write version comments

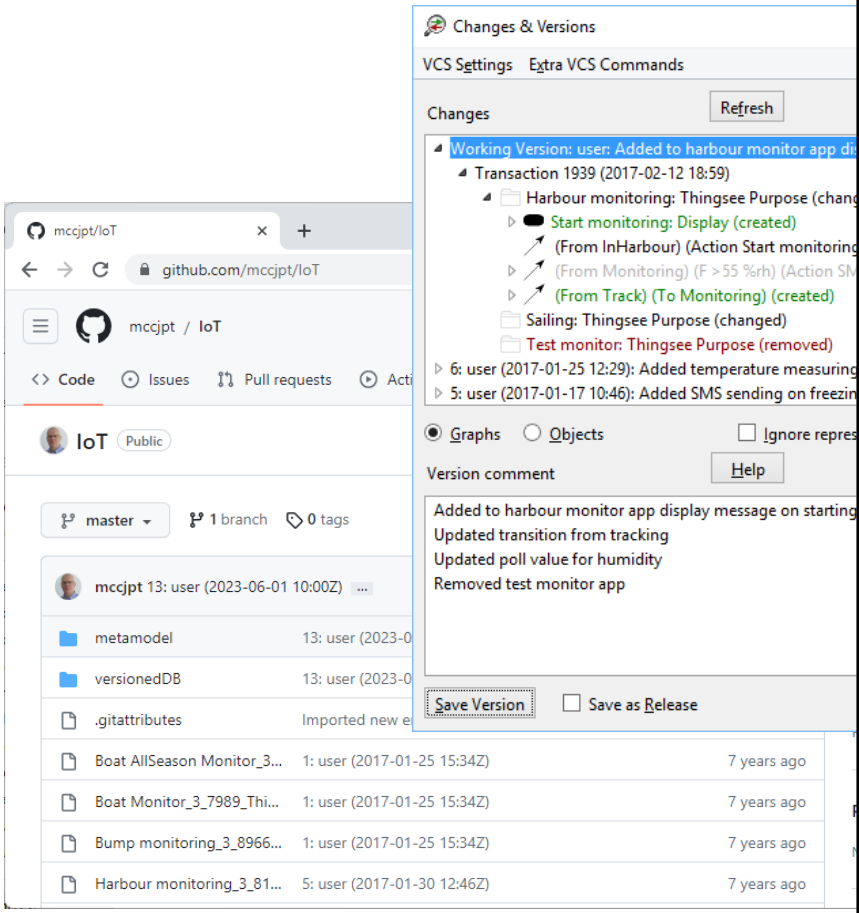


Compare changes in different ways



Versioning with external VCSs

- Simple setup and use
- Automated background execution of versioning commands
- Maintain same model in several places, syncing with GitHub etc.
- Supports multiple simultaneous versioners ...unlike some VCSs...
- Version and diff your language and generators along with models
- Git, SVN predefined
  - extend with your own



Task overview by role (in parallel)

<table>
<tr><th>Metamodeler</th><th>Layout Expert</th><th>Climate Specialist</th><th>Technical Modeler</th><th>Model Creator</th></tr>
<tr><td>Add new property slot</td><td rowspan="2">Rearrange diagrams:
- Sensors left
- Actions right</td><td rowspan="2">Update temperatures
- warmer, stabler
Update velocities
- rougher seas</td><td>Wait for metamodeler</td><td rowspan="2">Create new models:
- storm season
- tourist season</td></tr>
<tr><td>Add/update rules
Update symbols</td><td>Enter values for new property</td></tr>
</table>

We'll have one or more groups (Models1, Model2…)
and several roles in each group (meta1, layout1 in Models1…)

Quick demo…

Task overview by role (in parallel)

Metamodeler	Layout Expert	Climate Specialist	Technical Modeler	Model Creator
Add new property slot	Rearrange diagrams: - Sensors left - Actions right	Update temperatures - warmer, stabler Update velocities - rougher seas	*Wait for metamodeler*	Create new models: - storm season - tourist season
Add/update rules Update symbols			Enter values for new property	

We'll have one or more groups (Models1, Model2…)
and several roles in each group (meta1, layout1 in Models1…)

When we say you can ☺, here's the URL:
tinyurl.com/Models2023T2

MetaEdit+ gives you nothing

- No effort to install
- No effort to upgrade language & co-evolve models
- No effort to work together on same models
- No effort on copying or locking before editing
- No effort on merging after modeling
- No effort to move between PCs or networks
- No effort to get VCS version history
- No effort to support multiple languages
- No effort to upgrade to new tool versions
- Just model.

MetaCase

Thank you!

Questions? Experiences? Arguments?

Co-evolution paper: Tue 3, 09:00 W1
Co-evolution demo: Thu 5, 16:15 B6